```
class NumberFormat_Demo
{
    Public static void main (String args[])
    {
        try
        {
            int num = Integer.parseInt ("akki");
            System.out.println(num);
        }
        catch (NumberFormatException e)
        {
            System.out.println ("Number format exception");
        }
    }
}
```

output

Number format exception.

---

h) RuntimeException:

```
Public class ofG
{
    Public void Runtime ()
    {
        throw new Run();
    }
    Public static Void main (String [] args)
    {
        try
        {
```

```
        new GFG().Runtime();
        }
        catch(Exception x)
          {
           System.out.println("example of runtime exception");
          }
    }
}

    class Run extends RuntimeException
      {
        Public Run()
         {
          super();
         }
    }
```

Output:

example of runtime exception

---

i) IOException:

```
-> class IOException_Demo
     {
       Public static void main (String [] args)
        {
        Scanner scan = new Scanner ("Hello friends");
        System.out.println ("" + scan.nextLine ());
        System.out.println("Exception output :" scan.ioException ());
           scan.close();
     }
}
```

...o friends
...xception output : null

2] WAP to implement thread life cycle.

→ class thread implements Runnable

```
{
  Public void run()
  {
    try
    {
     Thread.sleep(1500);
    }
    catch (InterruptedException e)
    {
      e.printStackTrace();
    }
    System.out.println
    ("State of thread 1 while it called join() method on
        thread 2 " + Test.thread1.getState());

    try
    {
      thread.sleep(200);
    }
    catch(InterruptedException e)
    {
      e.printStackTrace();
    }
  }
}
```

```
Public static void main (String [] args)
  {
    obj = new Test();
    thread1 = new Thread(obj);
    System.out.println("State of thread 1 after creating it
                        + thread1.getState());
        thread1.start();

    System.out.println(" State of thread 1 after calling
      .start() method on it-" + thread1.getState());
  }

Public void run()
{
  thread myThread = new thread();
  Thread thread2 = new Thread(myThread);

  System.out.println("State of thread2 after creating it "
                      + thread2.getState());
                thread2.start();

  System.out.println("State of thread2 after calling .start
                      () method on it-" + thread2.get
                        State());

  try {
     Thread.sleep(200);
    }
```

```
catch (InterruptedException e){
        e.printStackTrace();
}

System.out.println("State of thread2 after calling .sleep()
                    method on it -"+thread2.getState
                    ());

try
  {
  thread2.join();
  }
  Catch(InterruptedException e)
    {
      e.printStackTrace();
    }
    System.out.println("State of thread2 when it has
      finished it's execution -"+thread2.getState());
  }
}
```

3] WAP to demonstrate multi-threading.

→
```
class MultithreadingDemo extends Thread
{
    Public void run()
    {
        try
        {
            System.out.println("Thread " + Thread.currentThread().
                                getId() + " is running");
        }
        catch (Exception e)
        {
            System.out.println("Exception is caught");
        }
    }
}

Public class Multithread {
    Public static void main (String [] args)
    {
        int n = 8;
        for(int i=0; i<n; i++)
        {
            MultithreadingDemo object = new Multithreading
                                        Demo();
            object.start();
        }
    }
}
```

88

eaad 15 is running
Thread 14 is running
Thread 16 is running
Thread 12 is running
Thread 11 is running
Thread 13 is running
Thread 18 is running
Thread 17 is running

4] WAP to demonstrate thread priority.

→
```
import java.lang.*;
class ThreadDemo extends Thread
  {
    Public void run()
      {
        System.out.println("Inside run method");
      }
    Public static void main (String [] args)
     ThreadDemo t1 = new ThreadDemo();
     ThreadDemo t2 = new ThreadDemo();
     ThreadDemo t3 = new ThreadDemo();

     System.out.println("t1 thread priority : "+t1.getPriority());
```

```
System.out.println("t2 thread priority : "+t2.getPrio
System.out.println("t3.thread priority: "+t3.getPriority
t1.setPriority(2);
t2.setPriority(5);
t3.setPriority(8);
System.out.println("t1 thread priority:"+t1.getPriority());
System.out.println("t2 thread priority :"+t2.getPriority());
System.out.println("t3 thread priority :"+t3.getPriority());
System.out.println("Currently Executing Thread :"
    +Thread.currentThread().getName()
    );
System.out.println("Main thread Priority :"
    +Thread.currentThread().getPriority
    ());
Thread.currentThread().setPriority(10);
System.out.println("Main thread Priority :"
    +Thread.currentThread().
    getPriority());
}
}
```

t1 thread Priority 5
t2 thread Priority 5
t3 thread Priority 5
t4 thread Priority 2
t2 thread Priority 5
t3 thread Priority 8

Currently Executing Thread : main
Main thread Priority : 5
Main thread Priority : 10

5] WAP to insert 10 student information into file.

→
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;

Public class abc
  {
    Public static void StudentInfo ()
    {
      JFrame f = new JFrame("Student Details Form");

    JLabel l1, l2, l3, l4, l5;
    JTextField t1, t2, t3;
    JcomboBox j1, j2;
    JButton b1, b2;
```

```
l1 = new JLabel("Student Name :");
l1.setBounds(50, 50, 100, 30);
l2 = new JLabel("college Email Id :");
l2.setBounds(50, 120, 120, 30);
l3. = new JLabel("Branch:");
l3.setBounds(50, 190, 50, 30);
l4 = new JLabel("Section:");
l4.setBounds(420, 50, 70, 30);
l5 = new JLabel("Mobile No:");
l5.setBounds(420, 120, 70, 30);

t1 = new JTextField();
t1.setBounds(150, 50, 130, 30);
t2 = new JTextField();
t2.setBounds(160, 120, 130, 30);
t3 = new JTextField();
t3.setBounds(490, 120, 130, 30);

String s1[] = {"", "CSE", "ECE", "EEE", "CIVIL",
               "MEC", "Others"};

String s2[] = {"", "Section-A", "Section-B", "Section-C",
               "Section-D", "Section-E"};
```

```
j1 = new JcomboBox(s1);
j1.setBounds(120,190,200,30);
j2 = new JcomboBox(s2);
j2.setBounds(420,50,160,30);
b1 = new JButton("Save");
b1.setBounds(150,300,70,30);
b2 = new JBoutton("close");
b2.setBounds(420,300,70,30);
b1.addActionListener(new ActionListener()
  {
    Public void actionPerformed(ActionEvent e)
    {
      String s1 = t1.getText();
      Sting s2 = t2.getText();
      String s3 = bj1.getSelected Item() + "\n";
      String s4 = j2.getselected Item() + "\n";
      String s5 = t3.get Text();
    if(e.getSource() == b1)
      {
      try
      { FileWriter w = new FileWriter("GFG.txt", true);
```

```
            kl.kWrite(s1+"\n");
            kl.kWrite(s2+"\n");
            kl.kWrite(s3+"\n");
            kl.kWrite(s4+"\n");
            kl.kWrite(s5+"\n");
            kl.close();
        }
        catch(Exception ae)
        {
            System.out.println(ae);
        }
    }
    JOptionPane.showMessageDialog(f,"successfully Saved"
                                  +"The Details");
    }
});
b2.addActionListener(new ActionListener()
{
  Public void actionPerformed(ActionEvent e)
  {
    f.dispose();
  }
});
```

```
f.addWindowListener (new WindowAdapter ()
  {
    Public void windowClosing (WindowEvent e)
      {
        System.exit (0);
      }
  });

  f.add (l1);
  f.add (t1);
  f.add (t2);
  f.add(t2);
  f.add (l3);
  f.add (j1);
  f.add (l4);
  f.add (j2);
  f.add (l5);
  f.add (t3);
  f.add(b1);
  f.add(b2);
  f.setLayout(null);
  f.setSize (700,600);
  f.setVisible (true);
}
public static void main (String args [])
  {
    StudentInfo ();
  }
}
```

Save

Close

] WAP to read the students information from a file.

```
import java.io.*;
Public class xyz
  {
    Public static void main (String args [])
    throws Exception
      {
        File file = new File ("c:\\Users\\Pankaj\\
                    Desktop\\test.txt");
        BufferedReader br = new BufferedReader (new
                            FileReader(file));
        String st;
        while ((st=br.readline()) != null)
        System.out.println (st);
      }
  }
```

output: If you want to code refer to JavaTpoint.

93

Name of the Programme - Practice Assignment - 6.java.

1] Write a program to implement default constructor.

```
→ Public class defaultConst
 {
   int rn;
   String name;
   defaultConst();
    {
      rn = 82;
      name = "Alone";
    }
  void display()
   {
     System.out.println(rn);
     System.out.println(name);
   }

Public static void main (String args[])
 {
   defaultConst a = new defaultConst();
   defaultConst b = new defaultConst();
   defaultConst c = new defaultConst();
 }
}
```

Q.2. Write a program to implement parameterized constructor.

→
```
Public class ParaConst
{
    int rn;
    String name;
    ParaConst(int x, String n)
    {
        rn = x;
        name = n;
        System.out.println(rn);
        System.out.println(name);
    }
    Public static void main (String args[])
    {
        ParaConst a = new ParaConst(10, "abc");
        ParaConst b = new ParaConst(20, "def");
        ParaConst c = new ParaConst(30, "ghi");
    }
}
```

) Write a program for constructor overloading. (Multiple constructor in a class)

```
Public class overloadConst
{
    int rn;
    String name;
    String add;
    int marks;
    overloadConst()
    {
        rn = 100;
        name = "khushi";
        add = "Lucknow";
        marks = 74;
```

```
overload Const(int x , String n, String a, int m)
{
    rn = x;
    name = n;
    add = a;
    marks = m;
}
Void see()
{
    System.out.println(rn);
    System.out.println(name);
    System.out.println(add);
    System.out.println(marks);
}
Public static void main(String[] args)
{
    overloadConst a = new overloadConst(20, "Alone", "Pune", 75);
    overloadConst b = new overloadConst();
    overloadConst c = new overloadConst(40, "sanu", "Sangola", 65);
    overloadConst d = new overloadConst();
    a.see();
    b.see();
    c.see();
    d.see();
}
```

q4. Write a program to implement single Inheritance concept showlers

→
```
class A
{
  int x=5;
}
class B extends A
{
  int y=5;
  void show()
  {
    int z = x*y;
    System.out.println("Multiplication is:="+z);
  }
}
Public class Sinherit
{
  Public static void main (String [] args)
  {
    B m = new B();
    m.show();
  }
}
```

Q.5: Write a program to show student information by using multilevel inheritance.

```
class Person
{
  String n;
  int a;
  void getPerson(String n1, int a1)
  {
    this.n = n1;
    this.a = a1;
  }
```

57

```
void showPerson()
  {
    System.out.println("student Name :-" +n);
    System.out.println("student Age :-" +a);
  }
}
class std extends Person {
      String stdId;
   void getStd(String stdid)
    {
      this.stdId = stdId;
   }
    void showStd()
      {
       System.out.println("student ID:-" +stdID);
      }
   }
   class clgStd extends std
      {
        String c;
        int y;
       void getClg(String c,int y)
        {
          this.c=c;
          this.y=y;
     }
      void show()
        {
          System.out.println("student Course:"+c);
          System.out.println("student year:"+y);
        }
   }
```

```
Public class std_info
{
   Public static void main (String [] a)
   {
      ClgStd stud = new ClgStd();
      stud.getPerson ("Sanika", 20);
      stud.getStd ("San1234");
      stud.getClg ("BCA-II", 2);
      stud.showPerson ();
      stud.showStd();
      stud.showClg ();
   }
}
```

3.6. Write a program to implement Hierarchical Inheritance.

```
class A
{
   int x = 100;
   int y= 50;
}
class B extends A
{
   void add()
   {
      int p = x+y;
      System.out.println ("Addition:-" +p);
   }
}
```

```
class c extends A
{
    void sub()
    {
        int q = x-y;
        System.out.println("Subtraction :-" + q);
    }
}

Public class HieInherit
{
    Public static void main(String [] args)
    {
        B m = new B();
        c n = new c();
        m.add();
        n.sub();
    }
}
```

9.7. Write a Program to implement Hybrid Inheritence.

```
class A
{
    int x = 100;
    int y = 20;
}

class B extends A
{
    void sub()
    {
        int p = x - y;
        System.out.println("Subtraction:-" + p);
    }
```

```
class C extends A
 {
   int z=50;
 }

class D extends C
   {
     void add()
     {
       int q = x+y+z;
       System.out.println("Addition -"+q);
     }
}

Public class Myb2Inherit
   {
     Public static void main (String [] q)
     B q = new B();
     D b = new D();
     a.sub();
      b.add();
   }
}
```

⑤ 8:: write a program to implement multiple inheritance using interface.

```
interface A
  {
    int a=10;
  }
interface B
  {
    int b=20;
  }
interface C
    { int c=30;
    } void call();
```

101

```
class D implements A,B,C
{
  public void cal()
  {
    int d=A.a+B.b+C.c;
    System.out.println("Addition :-"+d);
  }
}

Public class MultiInherit
  {
    Public static void main (String [] a)
      D obj=new D();
       obj.cal();
    }
  }
```

e.g. Write a program to store student information such as Rno, name, mark into student table (use type-4) driver.

```
// creating Table
Package jdbc;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;

Public class jdbc1
 {
   Public static void main (String [] a)
    {
      try
      {
        class.forName ("oracle.jdbc.driver.OracleDriver");
        connection con=DriverManager.getConnection("jdbc:
                                                    oracle:thin:@localhost
                                                    :1521:xe","system",
                                                    "password"); 102
```

```
        Statement stmt = con.createStatement();
        stmt.executeUpdate("create table stel(sno number, sname
                                varchar2(20), marks numb
        System.out.println("Table Created Successfully..");
        con.close();
    }
    catch(Exception e)
    {
        System.out.println("Run Time Error:: "+e);
    }
  }
}
```

// Inserting Record

```
Package jdbc;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
Public class jdbc2
 {
   Public static void main(String a[])
    {
     try
     {
      class.forName("oracle.jdbc.driver.OracleDriver");
      Connection con=DriverManager.getConnection("jdbc:oracle:
                                thin:@localhost:
                                1521:xe",
                                "system","password");
   PreparedStatement stmt = con.preparestatement("insert into stel
                                values(?,?,?)");
```

```
BufferedReader br = new BufferedReader (new InputStreamReader
                    (System.in));

while (true)
{
   System.out.println ("Enter Roll Number :->");
   int rno = Integer.parseInt (br.readline());
   System.out.println ("Enter student Name :->");
   String sname = br.readline();
   System.out.println ("Enter Marks :->");
   float marks = Float.parseFloat (br.readline());
   stmt.setInt (1, rno);
   stmt.setString (2, sname);
   stmt.setFloat (3, marks);
   int cnt = stmt.executeUpdate();
   System.out.println (cnt + "Record Inserted");
   System.out.println ("Do you want to Record Insert [Y/N]");
   String ch = br.readline();
   if (ch.equalsIgnoreCase ("N");
      break;
}
con.close();
}
catch (Exception e)
{
   System.out.println ("Run Time Error :-" + e);
}
}
```

10) Write a program to fetch student information from student table.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
Public class jdbc3
{
  Public static void main (String [] a)
  {
   try
   {
     class.forName("oracle.jdbc.driver.oracleDriver");
Connection con = DriverManager.getConnection("jdbc:oracle:thin
          :@localhost:1521:xe", "system", "password");
Statement smt = con.createStatement();
ResultSet rs = stmt.executeQuery("select * from std");
 while(rs.next())
  {
    System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "
                rs.getLong(3));
  }
  con.close();
}
catch (Exception e)
  {
   System.out.println("Run Time error" -"+e);
  }
 }
}
```

105

103